D212 Performance Assessment - Clustering Techniques

Name: Coots, Anthony.
Affiliation: Grad Student M.Sc Data Analytics.
Date: `2024-05-20`
Version: 1.5.2, r0.

Introduction

"In this task, you will act as an analyst and create a data mining report. You must select one of the data dictionary and data set files to use for your report from the following web link: "Data Sets and Associated Data Dictionaries.""

- WGU

Competencies

4030.6.4 : Clustering Techniques

- The graduate applies clustering techniques to accurately predict outcomes of interest.

Scenario

"One of the most critical factors in patient relationship management that directly affects a hospital's long-term cost-effectiveness is understanding the patients and the conditions leading to hospital admissions. When a hospital understands its patients' characteristics, it is better able to target treatment to patients, resulting in a more effective cost of care for the hospital in the long term.

You are an analyst for a hospital that wants to better understand the characteristics of its patients. You have been asked to use clustering techniques to analyze patient data to identify groups of patients with similar characteristics, ultimately enabling better business and strategic decision-making for the hospital."

- WGU

Table of Contents:

Research Question

A1: Proposal of Question

Propose one question that is relevant to a real-world organizational situation using k*-means (using only continuous variables) or hierarchical clustering*.

There is no single perfect question to raise in order to understand the patients and the conditions leading to hospital admissions as described in *Scenario*. However, by taking important pieces of data that describes the patients, such as their initial admission type, services they received while admitted and data like patient readmission, which means a patient has had been admitted again within thirty days of their initial admission, there exists a relevant question that aims to find insights to reduce patient readmission. Applying hierarchical clustering to the data to identify groups of patients based on their admit may help in identifying clusters, or groups of similar data points, for the services received. For each cluster, we can seek the rate of readmission of the given cluster and compare them among other clusters to see if certain groups have readmission risk. Investigating the differences in readmission rates in different admit types could be related to care they receive, specific policies or lack thereof. The structured question in order to facilitate this analysis is as follows:

Question: "*How do the types of initial admission (emergency, elective, observation admission) and the service received (blood work, CT scan, intravenous, MRI) relate to patient readmission*?"

A2: Defined Goal

Define one reasonable goal of the data analysis that is within the scope of the scenario and is represented in the available data.

The goal of this data analysis is to use hierarchical clustering to identify groups of patients categorized by their initial admission type (e.g., emergency, elective, observation) and services received (e.g., blood work, CT scan, intravenous, MRI). By defining these groups, the rate at which patients are readmitted in these groups can be discovered. By understanding these patterns/groups, interventions designed specific to the groups can address the needs for the group. This approach is to reduce readmission and strengthen the relationship between hospital and patient by improving patient care and implementing policy change better fit for the groups in need.

Technique Justification

B1: Explanation of the Clustering Technique

Logically explain how hierarchical clustering analyzes the 'medical_clean' *data set with expected outcomes.*

Hierarchical clustering is traditionally optimized for numerical data, relying on the calculation of distances between data points. For the data pertaining to the analysis in '*medical_clean*', which consists of categorical data like initial admission type (elective, emergency, observation) and service received (e.g., blood work, CT scan, intravenous, MRI), the technique still applies however looks a little different than usual to properly handle the categorical data.

Since hierarchical clustering traditionally uses numerical data, the metrics used to calculate the distance between data points changes. Usually these metrics are *Euclidean* or *Manhattan*. For the categorical data in this in analysis, utilizing '*Gower's*' Distance metric is sufficient as it is capable of handling binary, nominal or ordinal data values. A good article on '*Gower's*' Distance metric can be found here. This metrics calculates the distance based on the dimension of matching categories of the total number of data attributes.

Once the distances are calculated, the next step in the process involves using a linkage method to determine how the distances between the clusters (not data points) are calculated. Single linkage focuses on the shortest distance between calculations which is more sensitive to outliers as outliers are likely much farther. Complete linkage considers the longest distance, creates more balanced clusters. Other linkage methods exist like average or *Ward's* can also be considered.

For example, clusters might be different by initial admission type and may have different results for services received per the ten-thousand observations in the data set. This clustering gives way for exploration of patterns that are specific based on area and medical condition. These clusters may be more prone to patient readmission than others, a focal point of the analysis.

Ultimately, the expected outcomes is to identify the clusters that exhibit readmission and thus create interventions and policy adjustment aimed to reduce readmission for the groups that are at risk. Doing so may improve patient care and strengthen the relationship between hospital and patient.

B2: Summary of the Technique Assumption

Summarize one assumption of hierarchical clustering.

An assumption of hierarchical clustering is that data points closer together are more alike than data points that are farther apart. This concept is similar to observing that people who live close together in the same neighborhood are likely to have more in common comparatively to those who live in different areas. Similarly, data points that share certain characteristics are more likely to be grouped together.

B3: Packages or Libraries List

List the packages or libraries chosen for the analysis. Justify each item on the list.

gower

- Gower is a library used for the distance calculations for the categorical variables used in this analysis.

matplotlib.pyplot

- Library used for creating visualizations in Python. Used for dendrograms which help determine the number of clusters by assessing the generated tree structure visual.

numpy

- Library used for handling arrays and matrices, mathematical functions to operate arrays.

os

- Library used for navigating file paths, more specifically determining the current working directory.

seaborn

- Library used to build on top of matplotlib, works well with pandas DataFrames for less syntax and better plots.

scipy

- Library used for computing. For hierarchical clustering, using scipy.cluster.hierarchy for dendrogram and linkage classes and scipy.spatial.distance for creating dendrograms and computing distances, respectively.

sklearn

- Library used to assess the quality of the clusters created in the analysis.

warnings

- Library used to silence warnings as necessary such as new version warnings that do not affect the code.

In [19]:
```
pip install gower
```

```
Requirement already satisfied: gower in c:\users\acoots\appdata\local\anaconda3\lib
\site-packages (0.1.2)
Requirement already satisfied: numpy in c:\users\acoots\appdata\local\anaconda3\lib
\site-packages (from gower) (1.26.4)
Requirement already satisfied: scipy in c:\users\acoots\appdata\local\anaconda3\lib
\site-packages (from gower) (1.11.4)
Note: you may need to restart the kernel to use updated packages.
```

In [20]:
```
import gower
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
import seaborn as sns
from sklearn.metrics import silhouette_score
import warnings
```

Data Preparation

C1: Data Preprocessing

Describe one data preprocessing goal that is relevant to hierarchical clustering.

A data preprocessing goal for hierarchical clustering is to convert binary, nominal and ordinal data into a numerical format as hierarchical clustering requires numerical input to calculate distances between data points. For example, readmission (ReAdmis) in the data set contains values 'yes' or 'no' and should be coded as 1 and 0, respectively. Similarly, admission types and services areas such as elective and emergency or blood work and MRI need to be encoded via dummy variables to represent the existence (1) or not (0) of each category in a numerical format. This conversion is so that all data can be used in clustering calculations.

C2: Data Set Variables

Identify the data set variables needed to perform the analysis for the question. Accurately label each as continuous or categorical.

ReAdmis (categorical)

Initial_admin (categorical)

Services (categorical)

C3: Steps for Analysis

Explain each step used to prepare the data for analysis. Identify the code segment for each step.

Step 1: *Loading the data from file to DataFrame.*

Step 2: *Assess the data in the DataFrame for missing value and data types.*

Step 3: *If necessary, handle missing values/change data types.*

Step 4: *Remove features not in initial list.*

Step 5: *Encode categorical data.*

Step 1: *Loading the data from file to DataFrame.*

In [21]:
```
# What is my current working directory?
print("\n\n Current Working Directory: " + os.getcwd() + '\n')
```

```
 Current Working Directory: C:\Users\acoots\Desktop\Personal\Education\WGU\Data Ana
lytics, M.S\D212 - Data Mining II\Task 1 - Clustering Techniques
```

In [22]:
```
# Read data into DataFrame.
df = pd.read_csv("medical_clean.csv")
```

Step 2: *Assess the data in the DataFrame for missing value and data types.*

In [23]:
```
# Output the variables and the data types in the DataFrame.
summary = pd.DataFrame({
    "Variable": df.columns,
    "Missing Count": df.isna().sum(),
    "Data Type": df.dtypes}).reset_index(drop = True)

# Display summary of the DataFrame.
print(summary)
```

	Variable	Missing Count	Data Type
0	CaseOrder	0	int64
1	Customer_id	0	object
2	Interaction	0	object
3	UID	0	object
4	City	0	object
5	State	0	object
6	County	0	object
7	Zip	0	int64
8	Lat	0	float64
9	Lng	0	float64
10	Population	0	int64
11	Area	0	object
12	TimeZone	0	object
13	Job	0	object
14	Children	0	int64
15	Age	0	int64
16	Income	0	float64
17	Marital	0	object
18	Gender	0	object
19	ReAdmis	0	object
20	VitD_levels	0	float64
21	Doc_visits	0	int64
22	Full_meals_eaten	0	int64
23	vitD_supp	0	int64
24	Soft_drink	0	object
25	Initial_admin	0	object
26	HighBlood	0	object
27	Stroke	0	object
28	Complication_risk	0	object
29	Overweight	0	object
30	Arthritis	0	object
31	Diabetes	0	object
32	Hyperlipidemia	0	object
33	BackPain	0	object
34	Anxiety	0	object
35	Allergic_rhinitis	0	object
36	Reflux_esophagitis	0	object
37	Asthma	0	object
38	Services	0	object
39	Initial_days	0	float64
40	TotalCharge	0	float64
41	Additional_charges	0	float64
42	Item1	0	int64
43	Item2	0	int64
44	Item3	0	int64
45	Item4	0	int64
46	Item5	0	int64
47	Item6	0	int64
48	Item7	0	int64
49	Item8	0	int64

Step 3: *If necessary, handle missing values/change data types.*

There are no missing values. Data types will be changed for object variables upon encoding.

Step 4: *Remove features not in initial list.*

In [24]:
```
keepme = [
    'ReAdmis', 'Initial_admin', 'Services'
]

df_clean = df[keepme].copy()
```

In [25]:
```
summary = pd.DataFrame({
    "Variable": df_clean.columns,
    "Missing Count": df_clean.isna().sum(),
    "Data Type": df_clean.dtypes}).reset_index(drop = True)

# Display summary of the DataFrame.
print(summary)
```

```
        Variable  Missing Count Data Type
0        ReAdmis              0    object
1  Initial_admin              0    object
2       Services              0    object
```

Step 5: *Encode categorical data.*

In [26]:
```
# Categorical variables as yes or no.
non_dummy_nominal_variables = [
    'ReAdmis'
]

# Dummy variables.
dummy_variables = [
    'Initial_admin', 'Services'
]
```

In [27]:
```
for var in non_dummy_nominal_variables:
    boolean_int_dict = {str(var): {"No": 0, "Yes": 1}}
    df_clean.replace(boolean_int_dict, inplace = True)

# Pandas get_dummies method.
df_clean = pd.get_dummies(df_clean, columns = dummy_variables, drop_first = False)

for var in df_clean:
    if df_clean[var].dtype == "bool":
        df_clean[var] = df_clean[var].astype(int)
```

In [28]:
```
print(df_clean.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   ReAdmis                               10000 non-null  int64
 1   Initial_admin_Elective Admission      10000 non-null  int32
 2   Initial_admin_Emergency Admission     10000 non-null  int32
 3   Initial_admin_Observation Admission   10000 non-null  int32
 4   Services_Blood Work                   10000 non-null  int32
 5   Services_CT Scan                      10000 non-null  int32
 6   Services_Intravenous                  10000 non-null  int32
 7   Services_MRI                          10000 non-null  int32
dtypes: int32(7), int64(1)
memory usage: 351.7 KB
None
```

C4: Cleaned Data Set

Export an accurate copy of the cleaned data set.

In [29]:
```
# Export cleaned dataset to csv.
df_clean.to_csv("model_ready_medical_clean.csv", index = False)
```

Analysis

D1: Output and Intermediate Calculations

Determine the optimal number of clusters in the data set and accurately describe the methodology used.

The following code segment takes roughly 5-10 minutes **(Varies by machine specs.)**

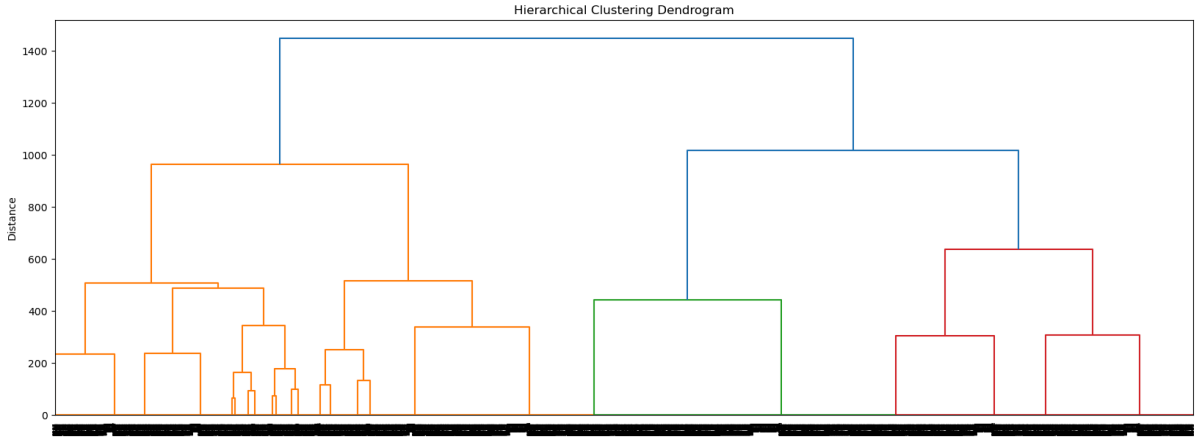
Last time log:

```
Kernel status: Idle
Executed 2 cells
Elapsed time: 249 seconds
```

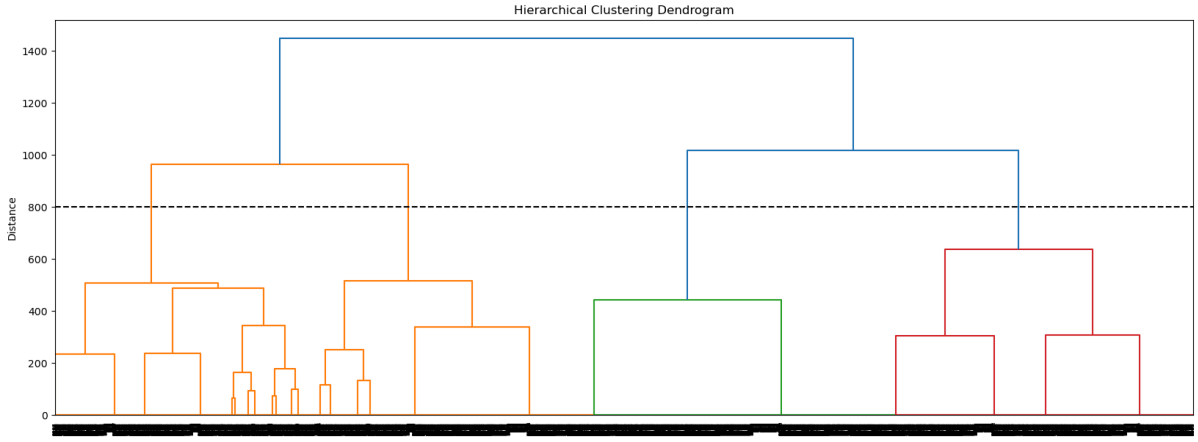
Specs: i9-11900k, 32 GB ram.

In [30]:
```
warnings.filterwarnings("ignore")

for var in df_clean:
    if df_clean[var].dtype == "int64":
        df_clean[var] = df_clean[var].astype(float)

# Perform hierarchical clustering
gowers_distance = gower.gower_matrix(df_clean)

linkage = linkage(gowers_distance, method = 'ward')
```

In [31]:
```
# Plot the dendrogram
plt.figure(figsize=(20, 7))
dendrogram(linkage)
plt.title('Hierarchical Clustering Dendrogram')
plt.ylabel('Distance')
plt.show()
```



In [32]:
```
# Plot the dendrogram
plt.figure(figsize=(20, 7))
dendrogram(linkage)
plt.title('Hierarchical Clustering Dendrogram')
plt.ylabel('Distance')
plt.axhline(y = 800, c = 'k', ls = '--', lw = 1.5)
plt.show()
```



Optimal number of clusters: 4

A dendrogram is used to find the optimal number of clusters for hierarchical clustering analysis. Each node represents a cluster, each branch connects the nodes where the length of the branch reflects the distance between the clusters. Each data point starts as its own cluster, then pairs of clusters are merged according to similarity.

Four seems like an optimal number of clusters given the dendrogram. Four, as seen by the intersecting points of the dashed line in the dendrogram, has a balance between too many small clusters and too few broad clusters. Additionally, the vertical distances seem to significantly reduce after this point. Determining the optimal number of clusters using Ward's method it is best practice the find the area that has the largest vertical distance without intersecting any horizontal lines. The intersection ultimately between the horizontal dashed lines and the vertical lines with the prior in mind, determines the optimal number of clusters.

D2: Code Execution

Provide the code used to perform the clustering analysis technique.

In [33]:
```
# Determine the number of clusters (e.g., based on the dendrogram.)
k = 4

cluster_labels = fcluster(linkage, k, criterion='maxclust')
df_cluster = df_clean.copy()
df_cluster['Cluster'] = cluster_labels

# Analyze clusters.
```

```
cluster_summary = df_cluster.groupby('Cluster').mean()
print(cluster_summary)
```

```
          ReAdmis  Initial_admin_Elective Admission  \
Cluster
1        0.361839                          0.510529
2        0.372508                          0.000000
3        0.379713                          0.000000
4        0.353234                          0.503636

         Initial_admin_Emergency Admission  \
Cluster
1                                      0.0
2                                      1.0
3                                      1.0
4                                      0.0

         Initial_admin_Observation Admission  Services_Blood Work  \
Cluster
1                                   0.489471                  0.0
2                                   0.000000                  0.0
3                                   0.000000                  1.0
4                                   0.496364                  1.0

         Services_CT Scan  Services_Intravenous  Services_MRI
Cluster
1                0.254835              0.668242      0.076923
2                0.262458              0.654070      0.083472
3                0.000000              0.000000      0.000000
4                0.000000              0.000000      0.000000
```

In [34]:
```
silhouette_scores = silhouette_score(df_cluster, cluster_labels)
print(" Silhouette Score: " + str(round(silhouette_scores, 3)))
```

```
Silhouette Score: 0.504
```

Data Summary and Implications

E1: Quality of the Clustering Technique

Logically explain the quality of the clustering technique.

In [35]:
```
silhouette_scores = silhouette_score(df_cluster, cluster_labels)
print(" Silhouette Score: " + str(round(silhouette_scores, 3)))
```

```
Silhouette Score: 0.504
```

The quality of the clusters in this analysis can be effectively assessed using a silhouette score, that evaluates both the compact nature of a cluster and the separation of the clusters on a scale of -1 to 1. Scores closer to 1 indicate that clusters are well-define and distinct, which is most ideal. Data points are closer matched to their own cluster and distant from neighboring clusters. Conversely, scores near -1 suggest improper assignment of data points, meaning that the data points are poorly matched to a cluster. A score closer to 0 indicates overlap among clusters.

The clusters in this analysis have a silhouette score of 0.504. This suggests that the clusters are fairly distinct and that the data points align well within their clusters, generally. Though there is room for improvement for peak separation (score of 1.) A score of 0.504 means that clusters are not overlapping inappropriately however they are not perfectly separate, either.

E2: Results and Implications

Discuss the results and implications of the clustering analysis.

The following lists are results for each cluster followed by their implications:

Cluster 1:

- Results:

Cluster 1 consisted of patient observations that had seldom counts of MRIs (~8%), some CT scans (~25%), and mainly involved intravenous services (~67%). About half of these patients (~51%) were admitted electively, while the other ~49% were admitted for observation. Of the patients in this cluster, 36% of them were readmitted within 30 days of discharge from either type of admission.

- Implications:

The readmission rate of this group while isn't a majority of patients, much too high to not act upon. This suggests a need for follow up post discharge for the patients receiving intravenous services. By doing such, the hospital(s) could see reduced patient readmissions.

Cluster 2:

- Results:

Cluster 2 had similar services as seen in cluster 1, with seldom MRI use, some CT scans and mostly intravenous services. However, all patients in this cluster were admitted through emergency services. Around 37% of these patients were readmitted within 30 days of their emergency admission.

- Implications:

Considering patients in this cluster were only of emergency admission and have a considerable readmission rate, there might exists a gap between emergency admission and the transition to and after discharge. Procedures that constitute a follow-up from the hospital in the best interest of the patient of their admission could address this gap and reduce patient readmission.

Cluster 3:

- Results:

Cluster 3 consisted of patients who exclusively received blood work and were admitted as an emergency. This population had a rate of readmission of approximately 38% within 30 days of initial admission, the highest rate of the 4 clusters.

- Implications:

This suggests that patients who had received blood work due to emergency conditions could have underlying health issues that have not been fully resolved by the time of discharge. Assessing policies in place for emergency admission may be helpful and introducing interventions to assess and manage these conditions could help reduce patient readmission.

Cluster 4:

- Results:

Cluster 4 was similar to cluster 3, such that patients had solely received blood work. These admissions were split between elective and observation admission. 35% of these patients were readmitted within 30 days.

- Implications:

The balance between elective and observation admissions and the considerable readmission rate should introduce review of the admission and discharge process for the hospital(s). Revising or creating new discharge care processes and investing in follow up care for these patients could help in reducing patient readmission.

E3: Limitation

Discuss one limitation of the data analysis.

One limitation of the analysis is the nature of the readmission data of the '*medical_clean*' data set. The data set provides the variable 'ReAdmis' as a binary variable to indicate whether or not a patient has been readmitted to the hospital within 30 days of their initial discharge. This however, does not indicate that the readmission was directly related to their previous admission or if the readmission was a separate incident; Readmission is only told as a true or false, yes or no value meaning there is no provided reason for exactly why the patient has been admitted to the hospital again and only that they were admitted again within 30 days of their prior "initial" admission.

E4: Course of Action

Recommend a reasonable course of action.

By examining how the types of admission and services received relate to patient readmission, as described in the question proposal, key interventions can be implemented from results and implications. Clusters 1 and 2 both see a high rate of intravenous service and considerable rate of readmission, suggesting the need for the creation or improvement of post discharge care. In order to do this, implementing phone calls, emails or visits home from the healthcare staff could support patients after their initial admission. Additionally, patient education programs could be helpful in educating patients of the conditions that lead to intravenous care, which together ultimately aims in reducing patient readmission.

Clusters 3 and 4 have exclusively involve blood work done as a service with a one cluster specific to emergency admissions and the other split between elective and observation admissions. This mix proposes the need to better the admission to discharge process. It is recommended to do further analysis in each of these cases of these clusters, insight may be uncovered on how admissions and care could be managed more effectively for more specific findings. These actions are designed to address specific challenges that were found in the clustering analysis, in order to reduce patient readmission and the associated fines.

Demonstration

Panopto Video of Programs

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6de465dd-2fb4-4359-a702-b1760139d656

Sources for Third-Party Code

2.3. Clustering. scikit learn. (n.d.). https://scikit-learn.org/stable/modules/clustering.html

DataCamp WGU Course Resource

gower 0.1.2. PyPI. (2022, November). https://pypi.org/project/gower/

warnings - Warning control. Python documentation. (n.d.). https://docs.python.org/3/library/warnings.html

Sources

Anand, D. (2024, February 20). Gower's distance. Medium. https://medium.com/analytics-vidhya/gowers-distance-899f9c4bd553

DataCamp WGU Course Resource

Liao, M., Li, Y., Kianifard, F., Obi, E., & Arcona, S. (2016, March 2). Cluster analysis and its application to healthcare claims data: A study of end-stage renal disease patients who initiated hemodialysis. BMC nephrology. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4776444/

Muhammad, U. S. (2023, June 9). Hierarchical clustering for categorical data. Medium. https://medium.com/@umarsmuhammed/hierarchical-clustering-for-categorical-data-168fe8fc0e2b